EZ Core is a framework designed to make creation of games within Unity easy and performance. EZ Core is designed to be as flexible and at the same time provide performance boost to your game as it minimizes unnecessary stuffs happening backend.

Actor

The Actor class is basically a normal MonoBehaviour class but it caches transform and gameObject references in it. Do not edit this class, new class should derive from here.

Action binders

The action binder class comes with Trigger & Collision for both 2D and 3D.

It is basically a component which does not need to be edited, just attach it to a gameObject with a trigger box in the scene and you can call a method from another class when the collision is detected. It is using UnityEvent. The same event used in uGUI OnClick().

(It mimics Unreal's LevelBlueprint.)

Good for scripted events, tutorials so you can make a script with all the scripted events and call them from all the triggers via action binders. Does not need to make so many scripts for each trigger anymore !

DebugClass

A wrapper around Unity Debug.Log, Debug.LogWarning, Debug.LogError.

Why and what's the difference ?

Being able to entirely shut down all the loggings with 1 step ☺

Comment out the #Defines to disable logging of that particular type.

EZ_Math

Collection of useful and common method that eases game creation.

A method that can get you a random item from a list of item with different probability values.

A list of useful explosion game logic codes.

The lists goes on and we do this because we don't want you to do the hard work.

EZ_PlayerPrefEx

Wrapper around Unity PlayerPref, Adds PlayerPref bool, vec2 and vec3.

EZ_Pooling_2

Pool Manager is a tool that manages instances that are needed over and over again in the same scene.

Objectives :

By managing a pool of instances and recycling objects that are not in use at that point of time, we can save on CPU overhead & memory that are required to instantiate & kill the objects.

So, pool manager can enhance performance when we are required to work with huge amount of objects that are needed frequently but only for a short amount of time.

Applications :

Spawning of bullets, particle effects, decals...

ActorPooled

The ActorPooled class is the same as the Actor class, with additional property that holds the pool name, and additional OnSpawned and OnDespawned method that is called automatically by the Pool Manager.

New classes must derive from this in order to use the pooling system.

To add new class, right click in project and Create -> NewActorPooled C#

It will auto generate the class and you just need to add in your custom stuff.

Pool Manager

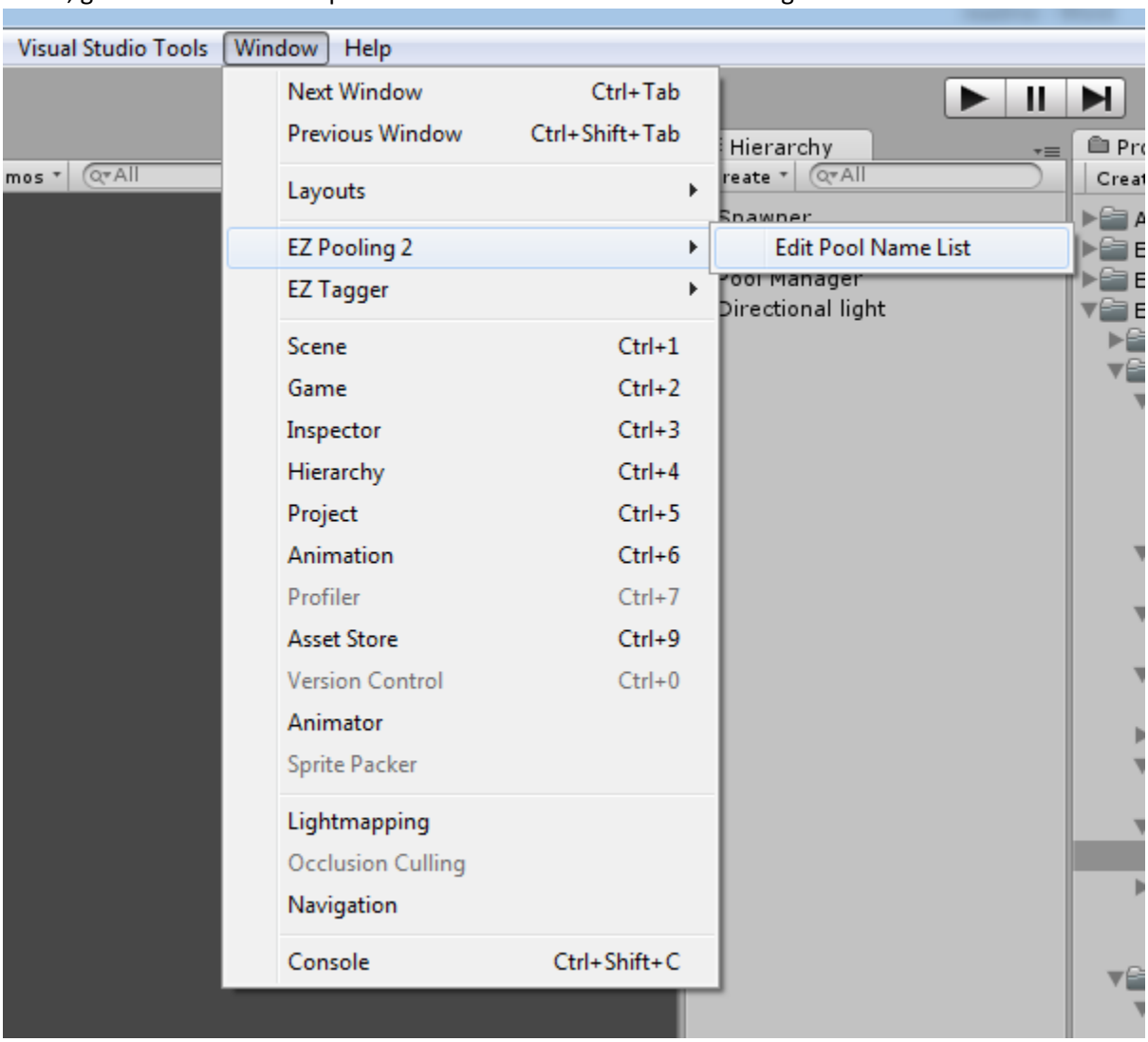
Must exist in the scene. It is the one which is responsible at handling all the object pools. Pools can either be predefined in the scene or be spawned when the 'Auto Add Missing Item' is enabled. Then the

newly spawned pool will take the default settings defined in the Pool Manager script, or uses its own via the PrefabPoolDefaultOverride.cs.

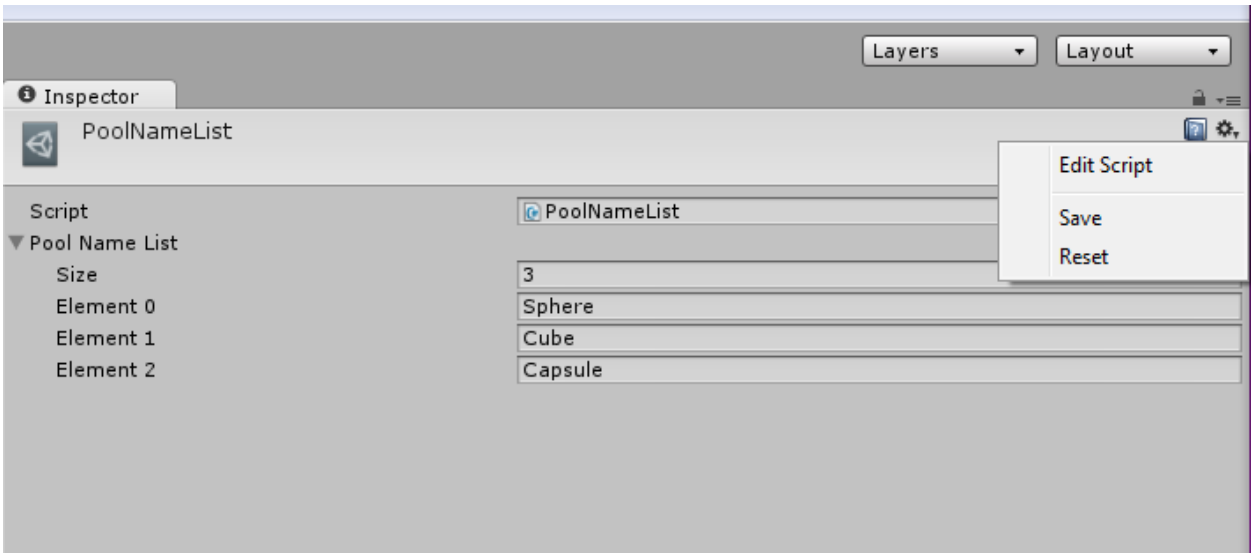
For users that have experienced EZ Pooling, this new pooling system now uses arrays to store the gameObject references, which is faster but loses the ability to grow.

How to use EZ Pooling 2 in your game ?

1. Create an empty gameObject and attach the pool manager script.
2. Make sure your prefabs has script that derive from ActorPooled.
3. Give your prefab a pool name. Cannot have similar name in the pool. If you need to add in more name, go to the Window drop down menu item and access EZ Pooling 2 -> Edit Pool Name List



4. After done adding, click the gear icon on the right and click Save.



5. After Unity finishes compiling, your new pool name should appear in the drop down property.
6. To Spawn, replace Instantiate(…) with PoolManager.Spawn(…)
7. To Despawn, replace Destroy(…) with PoolManager.Despawn(…)

EZ_Tagger

EZ Tagger is the multi-tagging version Unity tag. And it is faster and more efficient as it uses enum to compare instead of strings. It comes with 2 modes : Exact and Or.

Exact : the gameObject must have all the required tags in order to return true.

Or : at least 1 must match to return true.

To add more tag

1. To add more tag, go to the Window drop down menu item and access EZ Tagger -> Edit Tag List
2. After done adding, click the gear icon on the right and click Save.
3. After Unity finishes compiling, your new tag should appear in the drop down property.

FindGameObjectsWithTag

- The normal version behave like Unity FindGameObjectsWithTag
- The multitag version take in a list of tag and a Matcher type and return objects based on that rule.

Any issue or request for future version freely contact me at rudinesurya@gmail.com

Or you can follow this quick video tutorial

EZ Pooling 2 Setup vid

https://www.youtube.com/watch?v=94n1VjyRYOc

EZ Core Action Binder tutorial

https://www.youtube.com/watch?v=21WlJTVl0W0

EZ Tagger

https://www.youtube.com/watch?v=AWcBC3uP2go

https://www.youtube.com/watch?v=5WL4PQTTNdw